Amendments in the Claims

1. (currently amended) A computer-implemented method for identifying user interface (UI) objects in a markup-language stream, the method comprising the steps of:

receiving a predefined grammar for a particular application;

automatically generating a parser computer program based on the predefined grammar using an automated parser generator tool;

scanning ~~the~~ (i) the markup-language stream or (ii) a corresponding document object model (DOM) with the parser computer program to generate tokens;

parsing the tokens with the parser computer program to identify at least one UI objects in a portion of the particular application; and

outputting the portion of the particular application.

2. (original) The method of claim 1, wherein said markup-language stream drives a markup-language-based browser application, and wherein the scanning step includes scanning the DOM generated by a browser that displays that application.

3. (original) The method of claim 1, wherein the scanning step includes identifying elements of the DOM by traversal thereof.

4. (cancelled)

5. (previously presented) The method of claim 3, wherein the scanning step includes generating one or more tokens for each scanned DOM element.

6. (cancelled) .

7. (cancelled).

8. (currently amended) The method of claim 1~~-7~~, wherein the at least one UI objects comprises one of a user input fields, text fields, metatags, unprintable markup-language, or an in-line images.

9. (original) The method of claim 1, wherein the scanning and parsing steps are adapted to identify UI objects that correspond to elements displayed in the markup-language application.

10. (previously presented) The method of claim 1, further comprising grouping the tokens into syntactic structures that identify items displayed by the particular application.

11. (previously presented) The method of claim 10, wherein said step of grouping comprises identifying similarly formatted markup-language elements based on their markup-language attributes such as classname, font size, style, tag color, and size.

12. (previously presented) The method of claim 1, wherein said at least one UI objects comprises a name, content, a shape, or a location.

13. (previously presented) The method of claim 1, wherein automatically generating ~~said~~ the parser computer program comprises executing YACC ("Yet Another Compiler-Compiler").

14. (cancelled).

15. (cancelled).

16. (previously presented) The method of claim 1, wherein the parser computer program is a LALR(1) parser.

17. (currently amended) The method of claim 1, wherein the parser computer program is a LR(1) parser.

18. (previously presented) The method of claims 1, wherein the markup language is any of HTML, XHTML and XUL.

19. (currently amended) A ~~computer-readable medium encoded with computer program code, the computer program code~~ digital data processing system comprising:

a client digital data processor configured to:

~~program code for~~ receiving a predefined grammar for a particular application;

~~program code for~~ automatically generat~~e~~~~ing~~ a parser computer program based on the predefined grammar using an automated parser generator tool;

~~program code for~~ scan~~ning~~ the (i) the markup-language stream or (ii) a corresponding document object model (DOM) with the parser computer program to generate tokens;

~~program code for~~ pars~~e~~~~ing~~ the tokens with the parser computer program to identify at least one UI objects in a portion of the particular application; and

~~program code for~~ output~~ting~~ the portion of the particular application.

20. (currently amended) The digital data processing system ~~computer-readable medium~~ of claim 19, wherein the list of UI objects corresponds to elements displayed by the markup-language DOM.

21. (currently amended) The digital data processing system ~~computer-readable medium~~ of claim 20, wherein said UI objects comprise name, content, shape, location, and properties.

22. (cancelled).

23. (currently amended) The digital data processing system ~~computer-readable medium~~ of claim 19, wherein said tokens are interpreted according to the predefined grammar to identify and distinguish among UI objects of a markup-language application's display.

24. (currently amended) The digital data processing system ~~computer-readable medium~~ of claim 19, wherein the at least one UI object comprises a user input field~~s~~, a text field~~s~~, a metatag~~s~~, unprintable markup-language, or an in-line image~~s~~.

25. (currently amended) The digital data processing system ~~computer-readable medium~~ of claim 19, wherein the markup language is any of HTML, XHTML and XUL.

26. (previously presented) The method of claim 1, further comprising providing context-based help based at least in part on the portion of the particular application.

27. (currently amended) The digital data processing system ~~computer-readable medium~~ of claim 19, wherein the client digital data processor is further configured to ~~further comprising program code for~~ provid~~eing~~ context-based help based at least in part on the portion of the particular application.